

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

<hr/>)	
ILOG, INC., and ILOG, S.A.,)	
Plaintiffs,)	
)	
v.)	CIVIL ACTION
)	NO. 01-10648-WGY
BELL LOGIC, LLC,)	
Defendant.)	
<hr/>)	

MEMORANDUM

YOUNG, C.J.

January 9, 2002

ILOG, Inc., and ILOG, S.A. (collectively "ILOG"), brought this action seeking declaratory judgment, and Bell Logic fired back substantive counterclaims. After discovery closed, and prior to trial, ILOG brought this motion for summary judgment regarding, among other things, Bell Logic's claim for copyright infringement. This memorandum addresses ILOG's claim that it did not engage in copyright infringement.

I. Facts

ILOG is a computer software developer.¹ Def.'s Opp'n at 3. ILOG developed a "business rules" software suite called JRules. Pl.'s Mem. at 3. JRules is a computer program designed to codify and implement company-wide business practices (referred to as

¹ Software is used synonymously herein with program, computer software, or computer program.

"rules") through interfaces with numerous other software programs. Id. As an example of a rule, a hotel chain may have a policy of charging an extra fee per room when occupancy exceeds a certain percentage. JRules conceptually could interface with the accounting, reservation, and staffing systems to implement this policy on a company-wide scale.

Bell Logic and its predecessor LogicStore, LLC, developed a computer software program, called LogicStore, that runs in conjunction with JRules and, according to Bell Logic, enhances the features of JRules for its users. Def.'s Opp'n at 3. The features Bell Logic developed were seen as desirable by customers. Bell Logic alleges that ILOG formed a joint venture with it to design and market their complimentary products, id. at 3-4, and asserts that ILOG subsequently stole its ideas and products, integrated them into the next version of JRules, and then severed the relationship with Bell Logic. Id. at 4-7. Accordingly, Bell Logic seeks damages on a number of theories.

One of Bell Logic's theories is that, after it privately disclosed copyrightable components of its software to ILOG, ILOG infringed the Bell Logic copyrights by copying or incorporating elements of Bell Logic's software into ILOG's own revised version of JRules. Answer ¶¶ 20-28. Bell Logic identifies eighteen aspects of its product that it alleges are copyrightable, copyrighted, and copied in ILOG's latest version of JRules.

Def.'s Resp. to Pl.'s First Set of Interrogs. [hereinafter "Def.'s Resp."] Ex. 1.

Asserting that the eighteen aspects of LogicStore that it allegedly copied are not protected by the copyright laws, ILOG moves for summary judgment.

II. Analysis

A. Standard of Review

In determining a motion for summary judgment, the Court views the record "in the light most hospitable to the party opposing summary judgment, indulging all reasonable inferences in that party's favor." Griggs-Ryan v. Smith, 904 F.2d 112, 115 (1st Cir. 1990). "Summary judgment is warranted where the record, viewed in the light most favorable to the nonmoving party, reveals that there is no genuine factual dispute and the moving party [is] entitled to judgment as a matter of law." Siegal v. Am. Honda Motor Co., 921 F.2d 15, 17 (1st Cir. 1990); Fed. R. Civ. P. 56(c).

"The moving party is entitled to judgment as a matter of law if the nonmoving party does not adduce enough evidence to permit a reasonable trier of fact to find for the nonmoving party on any element essential to its claim." Milton v. Van Dorn Co., 961 F.2d 965, 969 (1st Cir. 1992). The party bearing the burden of proof must produce more than a "scintilla of evidence on each element essential to its claim, thus affording the jury a

nonconjectural basis for concluding that the fact to be inferred [is] more probable than its nonexistence." Id. (quoting Malave-Felix v. Volvo Car Corp., 946 F.2d 967, 970-71 (1st Cir. 1991)).

Rule 56(c) mandates the entry of summary judgment against a party who fails to demonstrate a genuine issue as to any material fact. Celotex Corp. v. Catrett, 477 U.S. 317, 322-23 (1986).

The "material" facts upon which the nonmovant relies to avoid summary judgment must reveal a genuine dispute "over facts that might affect the outcome of the suit under governing law."

Anderson v. Liberty Lobby, Inc., 477 U.S. 242, 248 (1986). The movant, however, is not required to make an affirmative showing that there are no material facts in issue. Instead, the movant only has to show an "absence of evidence to support the non-moving party's case." Celotex, 477 U.S. at 325. Furthermore, "[o]n issues where the nonmovant bears the ultimate burden of proof, he must present definite, competent evidence to rebut the motion." Mesnick v. General Elec. Co., 950 F.2d 816, 822 (1st Cir. 1991).

B. Analytic Framework

In order to prove copyright infringement, a plaintiff must show "(1) ownership of a valid copyright, and (2) copying of constituent elements of the work that are original." Feist Publications, Inc. v. Rural Tel. Serv. Co., 499 U.S. 340, 361 (1991). Software, in general, is amenable to copyright

protection. E.g., Lotus Dev. Corp. v. Borland Int'l, Inc., 49 F.3d 807, 813 n.5 (1st Cir. 1995) (citing 17 U.S.C. §§ 102(a)(1) (extending copyright protection to "literary works"), 101 (defining "literary works" to comprise "works . . . expressed in words, numbers, or other verbal or numerical symbols or indicia, regardless of the nature of the material objects, such as . . . disks . . . in which they are embodied.")). The Court therefore assumes that Bell Logic has a valid copyright in LogicStore as a whole, and that the first prong of Feist is satisfied. As will be shown later, however, this does not end the inquiry even under the first prong of Feist, as what matters in this case is not whether LogicStore as a whole is protected by copyright law, but whether the particular elements of LogicStore that Bell Logic alleges ILOG copied are protected.

Bell Logic must also show that ILOG copied the copyrightable elements of LogicStore. As the First Circuit observed in Lotus, there are two ways in which a plaintiff may proceed in satisfying the second prong of Feist:

[The plaintiff] may either present direct evidence of factual copying or, if that is unavailable, evidence that the alleged infringer had access to the copyrighted work and that the offending and copyrighted works are so similar that the court may infer that there was factual copying (i.e., probative similarity). The plaintiff must then prove that the copyrighting of copyrighted material was so extensive that it rendered the offending and copyrighted works substantially similar.

Lotus, 49 F.3d at 813 (internal citations omitted).

Application of this standard in the computer software field requires several steps. First, a distinction must be drawn between "literal" and "non-literal" copying. Id.; see also Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 701-702 (2d Cir. 1992) (Walker, J.). "Literal" copying is the verbatim copying of original expression, while "non-literal" copying is that which is paraphrased, or loosely paraphrased. Lotus, 49 F.3d at 814. Copyright protection is extended to non-literal copying "else a plagiarist would escape by immaterial variations." Nichols v. Universal Pictures Co., 45 F.2d 119, 121 (2d. Cir. 1930)(L. Hand, J.).

In something of a transmogrification of the terminology, this distinction has been extended to embrace different aspects of software programs. The actual source and object codes of a software program generally constitute the "literal" elements of a software program. Altai, 982 F.2d at 702.² "Non-literal" elements of a software program include its structure and fundamental essence. Id. at 701. This distinction reflects the idea that the non-literal copying of the source code is manifested by or through the non-literal elements. In addition:

It is important to differentiate between both literal

² Source code is the series of commands in a specific programming language that direct a computer to perform specific tasks (what is generally thought of as the program), object code is the binary language conversion of the source code (the machine readable code). See Altai, 982 F.2d at 698.

and nonliteral copying, as well as between literal and nonliteral elements of a computer program. The latter use of the terms 'literal' and 'nonliteral' is as terms of art, whereas the former is not. However, courts unfortunately often fail to distinguish between the two, or simply mischaracterize what is at issue in a certain case.

MiTek Holdings, Inc. v. Arce Eng'g Co., Inc., 89 F.3d 1548, 1556 n.16 (11th Cir. 1996).

In Lotus, the alleged copyright infringement involved spreadsheets -- Lotus's "1-2-3" and Borland's "Quattro Pro." 49 F.3d at 810. The issue in that case was whether Borland infringed Lotus's copyright in its "menu command hierarchy." Id. at 809. The menu command hierarchy is the system of pull down menus used to access and organize the user commands which operate the program. Id. Borland did not dispute that it "factually copied the words and arrangement of the Lotus menu command hierarchy," id. at 812, and both parties agreed that Borland did not copy the underlying source code. Id. at 810. Thus, in Lotus, the copying was not of the traditional literal elements of a software program, rather it was of what generally is referred to as the non-literal elements. Nevertheless, the First Circuit concluded that this was literal copying, id. at 814-15, thus obviating the need for any discussion of non-literal copying.

Non-literal copying, though unconsidered by the First Circuit, has been thoroughly addressed by the Second Circuit. In Altai, the Second Circuit considered the alleged infringement of

a program that enabled a computer program designed to operate with one operating system to operate with a different operating system. Altai, 982 F.2d at 698-99. Altai, the alleged infringer, at first copied the literal source code of the original program. Id. at 699-700. After the copyright suit was brought, Altai rewrote the appropriated code sections to achieve an identical functional result. Id. at 700. The Second Circuit opinion addressed whether Altai could be liable for copying the structure of the plaintiff's program, but not the actual code. The court, therefore, addressed the non-literal copying of a non-literal element. Id. at 702.

Interestingly, both Lotus and Altai considered copying of "non-literal" elements, that is, elements other than the source or object code. Lotus, 49 F.3d at 810; Altai, 982 F.2d at 702. Lotus, however, dealt with literal copying of non-literal elements, while Altai dealt with non-literal copying of non-literal elements. Compare Lotus, 49 F.3d at 815, with Altai, 982 F.2d at 702. Accordingly, literal copying of non-literal elements is governed by Lotus, and requires a preliminary analysis of whether the allegedly infringed components of the software program can be copyrighted. Lotus, 49 F.3d at 815.

With respect to non-literal copying, although the First Circuit has not adopted Altai, it has commented that "the Altai test may provide a useful framework for assessing the alleged

nonliteral copying of computer code" Lotus, 49 F.3d at 815. The Altai test requires a three step analysis: abstraction, filtration, and comparison. Altai, 982 F.2d at 706-10. The abstraction step requires courts to break a computer software program into its component parts and "isolate each level of abstraction within it." Id. at 707. As the First Circuit noted, "[t]his step enables courts to identify the appropriate framework within which to separate protectable expression from unprotected ideas." Lotus, 49 F.3d at 814. The second step, filtration, requires courts to determine whether the structural components identified in the first step are copyrightable. Id. (citing Altai, 982 F.2d at 707). The final step requires courts to "compare the protected elements of the infringed work (i.e., those that survived the filtration screening) to the corresponding elements of the allegedly infringing work to determine whether there was sufficient copying of protected material to constitute infringement." Lotus, 49 F.3d at 814 (citing Altai, 982 F.2d at 710); see also Recent Case, Computer Associates International, Inc. v. Altai, Inc., Nos. 91-7893, 91-7935, 1992 U.S. App. Lexis 14305 (2d Cir. June 22, 1992), 106 Harv. L. Rev. 510 (1992).

It is clear that this case involves non-literal elements. Interrogatory 1 of ILOG's first set of interrogatories required Bell Logic to "[i]dentify with particularity each and every item

(by which is meant . . . every line of code, every graphical feature, every function or attribute, etc.) in ILOG's JRules 3.0 software that was allegedly copied from Bell's LogicStore software," Pls.' First Set of Interrogs., no. 1. Bell Logic has identified eighteen areas of copying. Def.'s Resp. Ex. 1. None of the identified areas involves source or object code. Rather, each identifies what can, at best, be called a non-literal element of the LogicStore software. Accordingly, depending on whether literal or non-literal copying is alleged, Lotus or Altai will provide an appropriate analytic framework.

C. LogicStore's Copyrightable Elements

Bell Logic has not specified whether ILOG engaged in literal or non-literal copying, so the Court will examine the alleged copyright infringement using both a literal and a non-literal copying test.

1. Literal Copying

The guiding principle in determining whether material is properly protected by copyright laws is originality of expression. See, e.g., Baker v. Selden, 101 U.S. 99 (1879); Mazer v. Stein, 347 U.S. 201 (1954). At its core, copyright laws protect the original expression of ideas, and this constitutes the core distinction between copyrightable and noncopyrightable material: expression is protected, but the idea embodied by the expression is not. Thus, for example, while the King James Bible

could be protected, the idea of God, or the idea of a Bible, could not. This concept has been codified at section 102(b) of the Copyright Act:

In no case does copyright protection . . . extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work.

17 U.S.C. § 102(b). Recognizing this, Lotus requires a preliminary determination of whether the material copied was a method of operation, underlying idea, or the like, or whether the material copied was an expression. Lotus, 49 F.3d at 814.

Here, eighteen elements of two components of a software program are alleged to have been copied. These components constitute particular aspects of the business and developer's rule editors. In computer software parlance, "editor" refers to the interactive program sequence whereby users instruct the software to perform various functions. JRules, in essence, is designed to take certain business practices -- such as imposing surcharges upon young automobile renters -- and uniformly apply and integrate such rules into all the various software programs used by the company, such as accounting, inventory, reservations, and the like. The editor functions in JRules are used to create, edit and delete the various "rules" to be applied to other software. Def.'s Opp'n at 3.

In Lotus, the copied subject was a menu command hierarchy.

49 F.3d at 815. In Lotus, the First Circuit expressly embraced the concept of conducting the initial copyrightability test from an aggregate perspective. Id. Accordingly, the appropriate analysis does not break the putatively copyrightable work down into constituent components, but instead considers the work as a whole. Id. This Court therefore begins by analyzing Bell Logic's eighteen elements in the aggregate -- are rules editors themselves copyrightable? If they are not, then as in Lotus, there can be no infringement.³

In Lotus, the First Circuit stated that "'method of operation,' as that term is used in § 102(b), refers to the means by which a person operates something [I]f a new method of operation is used rather than described, other people would still be free to employ or describe that method." 49 F.3d at 815. The court concluded that the menu command hierarchy at issue was the actual method by which the program was operated or controlled. Id. at 816. It further concluded that there were multiple computer code formulations which would perform the same functions, lending credence to the conclusion that merely a method of operation was at issue. Id.

In this instance, the rules editors are the mechanism through which JRules is controlled -- the rules are created,

³ To the extent that the Court needs to examine the elements allegedly copied individually, it will do so in its consideration of non-literal copying under Altai.

edited, activated and deactivated. Def.'s Opp'n at 3; Pl.'s Mem. at 3; Def.'s Resp. Ex. 1, no. 1. Additionally, there is no allegation that the software code itself was copied, and there are certainly multiple methods by which source code could be written to achieve the same result -- two editors, for instance. Furthermore, if new rules editors were copyrightable, users of an older JRules version would still be able to make the computer program perform exactly the same as though they had the new editors, merely by using different keystrokes and rule entry procedures. In essence, the final capability of the program never changed; only the method of operating was altered.

As the First Circuit noted, "forcing the user to cause the computer to perform the same operation in a different way ignores Congress's direction in § 102(b) that 'methods of operation' are not copyrightable." Lotus, 49 F.3d at 818. Accordingly, the Court holds that rules editors are a method of operation, not an expression. The subject of the alleged infringement in this case, therefore, is a non-protected method of operation and cannot be the basis of any infringement.

The Lotus case involved "expression" coupled with a method of operation. There is no assertion that any expression of Bell Logic's was copied, however, and even assuming it was, such expression would not be copyrightable. In Lotus, the First Circuit rejected the district court's ruling that the menu

command hierarchy "constituted an 'expression' of the 'idea' of operating a computer program with commands arranged hierarchically into menus and submenus." 49 F.3d at 816. The court further rejected the assertion that mere expression created copyrightability, holding instead that where the expression is part of the method of operation, even the expression is not copyrightable. Id.

In this case, the Court is presented with a similar fact pattern. While Bell Logic may have made some expressive choices in developing its editors, the expressive choices are wholly embraced in the operation of the software program itself, and hence are unprotectable. Thus, to the extent that any of the particular areas of copying are asserted to be expressive, those areas are bound up in a method of operation that is not protected by the Copyright Act.

Lastly, the Court notes that its conclusion is consistent with Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435 (9th Cir. 1994), a case that considered the copyrightability of graphic user interfaces. In Apple, the Ninth Circuit considered whether the graphical system of manipulating icons, pull down windows, and the like -- once the defining characteristic of Apple computers -- could be replicated by Microsoft in its Windows operating system. Id. at 1443. In Apple, as here, the party alleging infringement submitted a list of particular

features in its computer program similar to those in the allegedly infringing work. Id. The Ninth Circuit affirmed the district court's decision to evaluate the list individually, and to rely on the list. Id. On the merits, the court concluded "Apple cannot get patent-like protection for the idea of a graphical user interface." Id. The user interface here, as in Apple, is not copyrightable.

2. Non-Literal Copying

The Court will analyze non-literal copying using the three-step Altai test discussed earlier. See supra p. 9. The abstraction step breaks the computer program down into its component parts, the filtration step determines if the components are copyrightable, and the comparison determines whether, after ignoring similarities due to those elements deemed uncopyrightable, substantial similarity exists between the programs. Altai, 982 F.2d at 706-10.

As a preliminary matter, the Court accepts the Altai test as the appropriate method for considering non-literal copying of non-literal elements of a software program. As mentioned earlier, while the First Circuit has not explicitly adopted Altai, it has suggested that the Altai analysis might be appropriate in the context of non-literal copying of non-literal elements of software. Lotus, 49 F.3d at 815. Furthermore, other circuits have adopted the Altai analysis or discussed it with

approval. E.g., MiTek Holdings, Inc. v. Arce Eng'g Co., Inc., 89 F.3d 1548, 1555-56 nn.15-16 (11th Cir. 1996) (commenting approvingly and adopting same test); Eng'g Dynamics, Inc. v. Structural Software, Inc., 26 F.3d 1335, 1343 (5th Cir. 1994) (adopting Altai's segmentation method); Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823, 834 (10th Cir. 1993) (adopting the abstraction-filtration-comparison test of Altai); see also Nimmer On Copyright § 13.03[F] (1985) (recommending the three-step test eventually adopted in Altai). Accordingly, the Court adopts the Altai test as the appropriate means by which to analyze non-literal copying.

Altai properly applied, however, ought be used to analyze an entire program or sub-program for alleged copying of its overall structure. The Lotus court contemplated, albeit in the literal copying context, an initial evaluation of the copyrightability of the program as a whole. It is this Court's opinion that an "aggregate evaluation" ought be a preliminary step to the Altai analysis in non-literal copying cases as well as literal copying cases.

The Court is not convinced that Altai, verbatim, is applicable except in the context of evaluating a claim that the overall structure, or look and feel, of an entire program has been copied. Altai arises in the context of demonstrating the second Feist prong -- substantial similarity of the works. The

principal difficulty with evaluating the substantial similarity of two entire programs is that the "similarity" may be due to "uncopyrightable similarities," not protected work which has been copied. Thus, Altai's principal usefulness lies in its identification of the constituent elements of a putatively copyrighted work, enabling separation of copyrightable expressions from noncopyrightable elements.

Here, this Court is presented not with allegations that an entire program has been copied, but rather with complaints that ILOG copied particular elements of a program. Thus, the Court is not faced with a need to separate the composite elements of an entire program into levels of abstraction; rather, the Court merely must determine whether the elements complained of are copyrightable. In Altai terms, the Court need not "abstract" the LogicStore program, it need only "filter" the identified elements.⁴

⁴ Judge Gorton dispensed with abstraction in Baystate Techs., Inc. v. Bentley Sys., Inc., but otherwise applied the Altai test where only certain elements of the computer program were allegedly copied. 946 F. Supp. 1079, 1089 (D. Mass. 1996). As this is a case of first impression in the First Circuit, the Court offers the following abstraction analysis in the interest of completeness:

Abstraction contemplates evaluation of copyrightability at the code level and each more generalized level of abstraction, ending with the program's general function. Altai, 982 F.2d at 707. The Altai test amounts to "reverse engineering" the computer program. See id.

[A] computer program can often be parsed into at least six levels of generally declining

abstraction: (i) the main purpose, (ii) the program structure or architecture, (iii) modules, (iv) algorithms and data structures, (v) source code, and (vi) object code. The main purpose of a program is a description of the program's function or what it is intended to do. When defining a program's main purpose, the court must take care to describe the program's function as specifically as possible without reference to the technical aspects of the program. The program's architecture or structure is a description of how the program operates in terms of its various functions, which are performed by discrete modules, and how each of these modules interact with each other. The architecture or structure of a program is often reduced to a flowchart, which a programmer uses visually to depict the inner workings of a program. Structure exists at nearly every level of a program and can be conceived of as including control flow, data flow, and substructure or nesting. Control flow is the sequence in which the modules perform their respective tasks. Data flow describes the movement of information through the program and the sequence with which it is operated on by the modules. Substructure or nesting describes the inner structure of a module whereby one module is subsumed within another and performs part of the second module's task.

The next level of abstraction consists of the modules. A module typically consists of two components: operations and data types. An operation identifies a particular result or set of actions that may be performed. For example, operations in a calculator program might include adding or printing data. A data type defines the type of item that an operator acts upon such as a student record or a daily balance. Algorithms and data structures are more specific manifestations of operations and data types, respectively. An algorithm is a specific series of steps that accomplish a particular operation. Data structure is a precise representation or specification of a data type that consists of (i) basic data type groupings such as integers or characters, (ii) values, (iii) variables, (iv)

arrays or groupings of the same data type, (v) records or groupings of different data types, and (vi) pointers or connections between records that set aside space to hold the record's values. The computer program is written first in a programming language, such as Pascal or Fortran, and then in a binary language consisting of zeros and ones. Source code is the literal text of a program's instructions written in a particular programming language. Object code is the literal text of a computer program written in a binary language through which the computer directly receives its instructions.

These generalized levels of abstraction will not, of course, fit all computer codes. Ordinarily, expert testimony will be helpful to organize a particular program into various levels of abstraction. In any event, as pointed out earlier, the organization of a program into abstraction levels is not an end in itself, but it is only a tool that facilitates the critical next step of filtering out unprotectable elements of the program.

Gates Rubber Co. v. Bando Chem. Indus., Ltd., 9 F.3d 823, 834-36 (10th Cir. 1993) (internal citations omitted).

LogicStore is designed to enhance certain features of the JRules software suite. As such, it focuses on improving several different aspects of the JRules software. One of these aspects is the rules editor module. The rules editor module is composed of various elements which represent user interface, rule editing functions, and display options. These individual elements are constructed using various data tables and algorithms, which in turn are constructed by the direct source and object codes. The Court, accordingly, identifies six levels of abstraction: object and source code, module elements (equivalent to algorithm and data values), modules, the program structure, and the overall program.

At the source and object code level, no copyrighting has occurred. See Def.'s Resp. Ex. 1, nos. 1-18 (citing only ideas as copied). Similarly, the overall structure of LogicStore has not been copied. See id. At the module level, copying is permitted because the overall module, as discussed supra, is a method of operation and hence not copyrightable. The text which follows addresses the rules editor (module) element level.

a. Bell Logic's Claims

In its answer to Interrogatory number one of ILOG's first set of interrogatories, Bell Logic alleges that ILOG copied eighteen aspects of LogicStore. Each aspect of copying Bell Logic identified constitutes a function element -- the basic building blocks of each function the program performs. Each of the allegedly copied aspects relates to the various rules editors. For each area identified below, Bell Logic asserts that JRules had never previously employed such a feature but, after being shown the features in LogicStore, ILOG incorporated each such feature into the next version of JRules.

The first alleged area of copying involves LogicStore employing two interactive "editors." LogicStore employed two alternate mechanisms: a technical editor for computer

programmers, and a basic editor for "business" users -- i.e. non-technical individuals who may not know computer programming.

Def.'s Resp. Ex. 1, no. 1. Apparently JRules previously had employed only one rules editor. Assuming this is true, the core question is whether using two editors (one technical and one not) is a copyrightable aspect of Bell Logic's software program.

Bell Logic identifies numerous aspects of its two-editor approach that it alleges ILOG copied into its newest version of JRules. Bell Logic alleges that ILOG copied its "free text editor model" from the developer's editor -- the technical rules editor for programmers; and from the business editor -- the easy-to-use editor for lay persons. LogicStore "implements a 'free-text' editing model wherein rule constructs are not fixed to pre-defined rigid anchors within the edit pane. This provides the user with an easy-to-read and easy-to-edit format. The user can edit clauses directly without first invoking an 'edit' or 'expand' mode." Def.'s Resp. Ex. 1, nos. 2, 11.

Second, Bell Logic asserts that ILOG copied its idea to have context-sensitive pop-up menus. This encompasses two ideas. First, LogicStore automatically presents -- "floating" above the cursor position -- a list of available commands or options. This is in contrast to the prior method, previously employed by ILOG, of having the potential commands fixed in a particular area of the editing pane. Next, LogicStore automatically adjusts the

contents of the list of available commands or options to reflect those appropriate to the portion of the rule being edited -- context sensitive pop-ups. This list of appropriate options automatically disappears once a selection has been made.

LogicStore's editors are also aware of the text being edited and its significance. Accordingly, LogicStore only suggests properties and methods (commands) appropriate to the text clause being edited. In LogicStore, when an operator "right-clicks" in a particular area of the editor pane, only choices appropriate to the portion being edited are displayed. ILOG allegedly copied this idea also, as its previous versions did not permit right-click menus at all.

ILOG's products were previously limited in their ability to recognize text developed outside the program. Apparently, the early JRules programs would not recognize certain characters in rules if the rule was created outside JRules and then imported, e.g., if created in a word processing program and then cut and pasted into JRules. As some rules may be quite long, Bell Logic thought it advisable to enable LogicStore to read and apply rules created in other text programs. After exposure to this idea, the next version of JRules was able to read imported rules.

LogicStore also implemented a design feature wherein the editor validated "not only the syntax of a rule but also ensure[d] that all values and objects in an expression are of the

appropriate type." Def.'s Resp. Ex. 1, nos. 8, 15. Apparently ILOG's products did not previously operate in this manner.

LogicStore also has a function which limits the availability of certain commands to users of the business editor model. The rationale for this limitation is that some functions are just too convoluted for lay persons to use, and ought be reserved for technicians. Furthermore, functions made available for the business users can be renamed for ease of lay person use.

Bell Logic also asserts that utilizing the business editor - the non-technical editor using lay terminology -- was copied by ILOG. It is difficult to see, however, how this is substantively different from Bell Logic's first alleged instance of copying.

LogicStore also uses "the mappings defined for JAVA constructs in Mapping Editor." Def.'s Resp. Ex. 1, no. 16. Previous versions of JRules did not, but the new version does.

LogicStore can also be used to "create domain specific implementations of rules based on the business objects. This implementation includes support for custom key words and internationalized keywords." Def.'s Resp. Ex. 1, no. 17. Previous versions of JRules did not, but the new version does.

LogicStore also began to color code keywords, in order for operators to identify classes and differentiate them from data values more easily. This facilitated rule creation, editing, and activating the various rules. JRules had previously not done

this, but the new version does.

The copying alleged by Bell Logic in each of the above instances is copying of an idea -- an idea which will permit a software user to edit or create rules in a certain manner. Ideas -- for example, using two editors instead of one, having a free text editor, color coding keywords -- are simply not copyrightable. If the alleged copying was of the screen displays, then perhaps expression would be involved. The copying alleged, however, is copying of the ideas themselves. ILOG's copying, even if true, cannot constitute infringement of Bell Logic's copyright in LogicStore as a whole.

Bell Logic also asserts ILOG copied the actual mechanism through which the program is controlled. Improvements in the method of operation of the program, whether through pop-up menus in certain situations, free text editors, or context-sensitive program suggestions, are nothing more than expressions of methods of operation and, as the First Circuit made clear in Lotus, expression of methods of operation is simply not copyrightable. Lotus, 49 F.3d at 816. There are numerous methods of writing the source code so as to achieve identical functions, which supports the proposition that what is at issue here is merely the method of operation, not the expression. See Lotus, 49 F.3d at 815.

Having determined that none of the elements of the LogicStore program allegedly copied are susceptible of copyright

protection, none can form the basis for the comparison required in the third Altai step. That is, substantial similarity between the allegedly infringing and infringed works cannot be demonstrated using these elements. Because the Court has already held that a rules editor is an uncopyrightable method of operation, the Court concludes that the rules editor program as a whole cannot be the basis for comparison.

Thus, ILOG did not infringe on any copyrightable material contained in LogicStore.

III. Conclusion

This aspect of the case boils down to a single important question: Is the material allegedly copied protected by copyright? The Court answers the question in the negative.

Individually, the elements are noncopyrightable ideas. In the aggregate, the elements constitute a method of operation. Either way, Bell Logic is not entitled to copyright protection. Accordingly, ILOG's motion for summary judgment [Docket No. 22] was GRANTED on December 20, 2001, with respect to the copyright claims [Docket No. 31].

WILLIAM G. YOUNG
CHIEF JUDGE

Publisher Information

**Note* This page is not part of the opinion as entered by the court.
The docket information provided on this page is for the benefit
of publishers of these opinions.**

U.S. District Court - Massachusetts (Boston)

CIVIL DOCKET FOR CASE #: 01-CV-10648

Ilog, Inc., et al v. Bell Logic, LLC Filed: 04/19/01
Assigned to: Chief Judge William G. Young Jury demand: Defendant
Demand: \$0,000 Nature of Suit: 820
Lead Docket: None Jurisdiction: Federal Question
Dkt # in California Dist. Crt : is C00-21228

ILOG, INC. William S. Eggeling
Plaintiff

Catherine A. Mondell

Ropes & Gray
One International Place
Boston, MA 02110
617-951-7000

ILOG, INC. William S. Eggeling
Plaintiff

Catherine A. Mondell

Ropes & Gray
One International Place
Boston, MA 02110
617-951-7000

Gray, Cary, Ware & Freidenrich
1755 Embarcadero Road
Palo Alto, CA 94303-3340
650-833-2000

Gray, Cary, Ware & Freidenrich
1755 Embarcadero Road
Palo Alto, CA 94303-3340
650-833-2000

Gray, Cary, Ware & Freidenrich
1755 Embarcadero Road
Palo Alto, CA 94303-3340
650-833-2000

ILOG, SA William S. Eggeling

Plaintiff (See above)

Catherine A. Mondell
(See above)

ILOG, SA
Plaintiff William S. Eggeling
(See above)

Catherine A. Mondell
(See above)

BELL LOGIC, LLC
Defendant

Emanuel D. Torti

Torti & Visnick
144 Main Street
Norfolk, MA 02056
508-541-4100

John J. Flanagan

Torti Flanagan
146 Main Street
Norfolk, MA 02056
508-541-4100

Berliner, Cohen
Ten Almaden Blvf. Ste. 1100
San Jose, CA 95113
408-286-5800

Berliner, Cohen
Ten Almaden Blvf. Ste. 1100
San Jose, CA 95113
408-286-5800

BELL LOGIC, LLC

Berliner, Cohen
Ten Almaden Blvf. Ste. 1100
San Jose, CA 95113
408-286-5800

BELL LOGIC, LLC

Berliner, Cohen
Ten Almaden Blvf. Ste. 1100
San Jose, CA 95113
408-286-5800

Counter-Defendant

Gray, Cary, Ware & Freidenrich
1755 Embarcadero Road
Palo Alto, CA 94303-3340
650-833-2000

Counter-Defendant

Gray, Cary, Ware & Freidenrich
1755 Embarcadero Road
Palo Alto, CA 94303-3340
650-833-2000

Counter-Defendant

Gray, Cary, Ware & Freidenrich
1755 Embarcadero Road
Palo Alto, CA 94303-3340
650-833-2000

ILOG, SA

ILOG, SA

ILOG, SA

BELL LOGIC, LLC
Counter-Claimant

Emanuel D. Torti

Torti & Visnick
144 Main Street
Norfolk, MA 02056
508-541-4100

Torti Flanagan

Torti Flanagan
146 Main Street
Norfolk, MA 02056
508-541-4100

Berliner, Cohen
Ten Almaden Blvf. Ste. 1100
San Jose, CA 95113

Berliner, Cohen
Ten Almaden Blvf. Ste. 1100
San Jose, CA 95113

ILOG, INC. William S. Eggeling
Counter-Defendant Catherine A. Mondell

Ropes & Gray
One International Place
Boston, MA 02110
617-951-7000

ILOG, INC. William S. Eggeling
Counter-Defendant Catherine A. Mondell

Ropes & Gray
One International Place
Boston, MA 02110
617-951-7000

Gray, Cary, Ware & Freidenrich
1755 Embarcadero Road
Palo Alto, CA 94303-3340
650-833-2000

Gray, Cary, Ware & Freidenrich
1755 Embarcadero Road
Palo Alto, CA 94303-3340
650-833-2000

Gray, Cary, Ware & Freidenrich
1755 Embarcadero Road
Palo Alto, CA 94303-3340
650-833-2000

ILOG, SA William S. Eggeling
Counter-Defendant (See above)

Catherine A. Mondell
(See above)

ILOG, SA William S. Eggeling
Counter-Defendant (See above)

Catherine A. Mondell
(See above)